Competitive Programming (SoC'25)

Project id - 22

Mentor - Himanshu Shete(23B0770)

## Week 2 : Strings, Pattern Matching, Recursion, **Number Theory** (GCD, Sieve, Modular Arithmetic)

### Theory:

(these are just resources you can always learn from youtube or other sources)

1. Strings
   1. <https://www.geeksforgeeks.org/strings-in-cpp/>
   2. <https://cplusplus.com/reference/string/string/>
2. Pattern matching
   1. <https://www.geeksforgeeks.org/kmp-algorithm-for-pattern-searching/>
   2. <https://cp-algorithms.com/string/z-function.html>
3. Recursion
   1. <https://www.geeksforgeeks.org/introduction-to-recursion-2/>
   2. <https://www.geeksforgeeks.org/introduction-to-backtracking-2/>
4. Number Theory
   1. <https://www.geeksforgeeks.org/modular-arithmetic-for-competitive-programming/>
   2. <https://codeforces.com/blog/entry/72527>
   3. <https://cp-algorithms.com/algebra/module-inverse.html>
   4. <https://cp-algorithms.com/algebra/sieve-of-eratosthenes.html>
   5. <https://cp-algorithms.com/algebra/factorization.html>
   6. <https://cp-algorithms.com/algebra/binary-exp.html>
   7. <https://www.w3schools.com/dsa/dsa_ref_euclidean_algorithm.php>
   8. <https://cp-algorithms.com/algebra/euclid-algorithm.html>
   9. <https://cp-algorithms.com/algebra/extended-euclid-algorithm.html>
   10. <https://cp-algorithms.com/algebra/linear-diophantine-equation.html>

### Problems:

(increasing difficulty, maintain a git repo)

1. Strings and Pattern matching
   1. <https://www.codechef.com/practice/course/strings/STRINGS/problems/DNASTORAGE> (easy)
   2. <https://codeforces.com/problemset/problem/1155/A>
   3. <https://codeforces.com/problemset/problem/1374/C>
   4. <https://leetcode.com/problems/reverse-words-in-a-string/description/>
   5. <https://codeforces.com/problemset/problem/1506/C>
   6. Kmp- <https://www.spoj.com/problems/NHAY/>
   7. <https://www.codechef.com/problems/FCTRL2> (yes! Its strings)
   8. Z- <https://codeforces.com/contest/126/problem/B>
2. Recursion and Backtracking
   1. <https://leetcode.com/problems/subsets/description/> (classic)
   2. <https://leetcode.com/problems/permutations/description/>
   3. <https://leetcode.com/problems/n-queens/description/>
   4. <https://leetcode.com/problems/sudoku-solver/description/>
3. Number Theory

Gcd

* 1. <https://codeforces.com/problemset/problem/1498/A>
  2. <https://codeforces.com/problemset/problem/1543/A>
  3. <https://cses.fi/problemset/task/1081/>
  4. <https://codeforces.com/problemset/problem/1183/B>
  5. <https://www.codechef.com/practice/course/number-theory/INTNT01/problems/STRNG>
  6. <https://codeforces.com/problemset/problem/1872/C>
  7. <https://codeforces.com/problemset/problem/633/B>

Modular

1. <https://codeforces.com/problemset/problem/913/A>
2. <https://cses.fi/problemset/task/1095/>
3. <https://codeforces.com/problemset/problem/1285/A>

Sieve of eratosthenes(prime numbers)

1. <https://cses.fi/problemset/task/2182/>
2. <https://www.spoj.com/problems/PRIME1/>
3. <https://codeforces.com/contest/26/problem/A>
4. <https://codeforces.com/problemset/problem/17/A>
5. <https://codeforces.com/contest/776/problem/B>
6. <https://www.hackerrank.com/contests/projecteuler/challenges/euler134/problem>
7. <https://projecteuler.net/problem=146>
8. <https://codeforces.com/problemset/problem/154/B>

Diophantine- <https://codeforces.com/problemset/problem/7/C>